

Rule
1.126

- 23/ 3. A method as claimed in claim ²²2 further comprising retrieving all linked data structures in the thread stack.
- 24/ 4. A method as claimed in claim ²³3 further comprising retrieving a last data structure having no next data structure reference.
- 25/ 5. A method as claimed in any of the preceding claims further comprising retrieving the first object data structure in the thread stack referenced by a first object data structure pointer.
- 26/ 6. A method as claimed in claim ²⁵5 comprising:
defining a reachable set of objects as all objects referenced directly or indirectly by the root set objects.
- 27/ 7. A method as claimed in claim ²⁶6 further comprising identifying all objects within the process and reclaiming the memory space of all non-reachable objects.
- 28/ 8. A method as claimed in claim ²⁶6 or ²⁷7 further comprising moving reachable objects so that they are contiguous in memory and updating all object references in the thread stack by tracing through the chain of object data structures.
- 29/ 9. A method of managing an object in a thread stack based garbage collected virtual machine comprising:
storing an object data structure in the thread stack comprising a reference to the object and a reference to a previously stored object data structure in the stack;
whereby the object data structure and the previously stored object data structure form a root set of data object structures.
- 30/ 10. A method as claimed in claim ²⁹9 further comprising linking the object data structure to the previously stored object data structure.

Rule
1.126

31 11. A method as claimed in claim ³⁰11 further comprising:
storing a variable pointing to the previously stored object data structure at the top
of the stack;

using the variable when storing a new object data structure; and
updating the variable with the new object data structure reference.

32 12. A method as claimed in claim ³¹11 further comprising:
saving the variable pointer;
storing the object data structure;
updating the variable with the reference to the latest stored object data structure;
performing the process ; and
restoring the stack pointer.

33 13. A method as claimed in claim ²⁹9 further comprising:
retrieving an object data structure and extracting the associated object reference
and data structure reference;
using the associated data structure reference to retrieve the previously stored
object data structure;
retrieving all the object references in the stack by tracing through the chained of
object data structures.

34 14. A method as claimed in claim ³³13 comprising:
identifying all objects referenced directly or indirectly by the root set objects and
marking the root set and all referenced objects as reachable.

35 15. A method as claimed in claim ³⁴14 further comprising identifying all objects within
the process and reclaiming the memory space of all non-reachable objects.

36 16. A method as claimed in claim ³⁵15 further comprising moving reachable objects in
process memory so that they are contiguous and updating all object references in the
stack by tracing through the chain of object data structures.